Pergamon

Appl. Math. Lett. Vol. 8, No. 4, pp. 79–82, 1995

0893-9659(95)00052-6
0893-9659/95 $9.50 + 0.00

Numerical Solution of Differential and Algebraic Equations for a Flow Model with Diffusion

K. SUN AND R. P. TEWARSON
Department of Applied Mathematics and Statistics
State University of New York at Stony Brook
Stony Brook, NY 11794-3600, U.S.A.

A. M. WEINSTEIN AND J. L. STEPHENSON
Department of Physiology, Cornell University Medical College
1300 York Avenue, New York, NY 10027, U.S.A.

(Received November 1994; accepted January 1995)

Abstract—Two algorithms are described for solving systems of differential and algebraic equations arising in mathematical models of solute and water flow along a tube with permeable walls. Some computational results comparing the two algorithms are presented.

Keywords—Mathematical models, Ordinary differential equations, Numerical solutions.

1. INTRODUCTION

We consider the numerical solution of the following differential and algebraic equations:

$$\frac{dy}{dx} + f(y,z) = 0, \tag{1}$$

$$A(y,z) = 0 \tag{2}$$

and

$$\frac{dU(y,z,\alpha z')}{dx} + V(y,z) = 0 \tag{3}$$

for $y(x)$ and $z(x)$, where $0 \le x \le 0.5$, and $y(0)$ and $z(0)$ are given. This problem arises in a segmental model of the proximal tubule of the mammalian kidney. The problem has been solved numerically for $\alpha = 0$ by Weinstein [1,2]. The term $\alpha z'$ is necessary if diffusive flow is included in the model—the diffusion coefficient $\alpha = 10^{-5}$. We plan to include such detailed segmental tubular models in our medullary and whole kidney models. We have found that one-step methods based on the Trapezoidal Rule are extremely well suited for our models [3,4]. Weinstein [1,2] has also used such schemes for solving (1)–(3) for $\alpha = 0$ with $y, f \in \mathbb{R}^2$, $A \in \mathbb{R}^{34}$, $U, V \in \mathbb{R}^{10}$ and $z \in \mathbb{R}^{44}$. The second- or fourth-order predictor-corrector schemes turned out to be unsuitable for our models due to convergence and stability problems.

Research supported by NIH grants DK1759314, DK29857, DK 31550, RR 06589 and NSF Grant DMS9216614.

Typeset by $\mathcal{A}\!\mathcal{M}\!\mathcal{S}$-TEX

2. NUMERICAL SOLUTION

Let n be the number of subintervals into which the x range is subdivided. Let $h = 0.5/n$, $x_j = (j-1)h$, $y_j = y(x_j)$ and $z_j = z(x_j)$; where $j = 1, 2, \dots, n+1$. Then integrating (1) and (3) from x_{j-1} to x_j and evaluating (2) at x_j, we have

$$y_j - y_{j-1} + \frac{h}{2}\left[f\left(y_{j-1}, z_{j-1}\right) + f\left(y_j, z_j\right)\right] = 0, \tag{4}$$

$$A\left(y_j, z_j\right) = 0 \tag{5}$$

and

$$U\left(y_j, z_j, \alpha z_j'\right) - U\left(y_{j-1}, z_{j-1}, \alpha z_{j-1}'\right) + \frac{h}{2}\left[V\left(y_{j-1}, z_{j-1}\right) + V\left(y_j, z_j\right)\right] = 0. \tag{6}$$

Equations (4) and (6) have $\mathrm{O}(h^3)$ discretization errors. In (6), we replace z_{j-1}' and z_j' with numerical derivatives having $\mathrm{O}(h^3)$ errors. We include the $\alpha z'$ terms in the so-called deferred correction mode in the following algorithms. Since α is small, both algorithms converge rapidly.

ALGORITHM 1. Given y_1 and z_1.

- Step 1: Set $\alpha = 0$.
- Step 2: Compute y_j, z_j for $j = 2, 3, 4$ by solving the system of nonlinear equations (4)–(6) for y_j, z_j using Newton's Method.
- Step 3: Compute z_j' from the z_j's computed in Step 2 by using the following numerical derivatives [5, pp. 554–557].

$$z_1' = \frac{1}{6h}\left(-11z_1 + 18z_2 - 9z_3 + 2z_4\right). \tag{7}$$

Since we have no information about z_1', we use the above formula as suggested in [6, p. 56],

$$z_2' = \frac{1}{6h}\left(-2z_1 - 3z_2 + 6z_3 - z_4\right), \tag{8}$$

$$z_3' = \frac{1}{6h}\left(z_1 - 6z_2 + 3z_3 + 2z_4\right) \tag{9}$$

and

$$z_4' = \frac{1}{6h}\left(-2z_1 + 9z_2 - 18z_3 + 11z_4\right). \tag{10}$$

Set $\alpha = 10^{-5}$ and using the most recently computed values of z_j', repeat Steps 2 and 3 until the changes in z_j' values for two consecutive iterations are small.

Now for $5 \le j \le n+1$, as initial guesses for y_j, z_j and z_j' use the following:

- Step 4:

$$y_j = 4y_{j-1} - 6y_{j-2} + 4y_{j-3} - y_{j-4}, \tag{11}$$

$$z_j = 4z_{j-1} - 6z_{j-2} + 4z_{j-3} - z_{j-4}, \tag{12}$$

$$z_j' = \frac{1}{6h}\left(26z_{j-1} - 57z_{j-2} + 42z_{j-3} - 11z_{j-4}\right). \tag{13}$$

- Step 5: Solve (4)–(6) for y_j, z_j using Newton's Method.
- Step 6: Compute

$$z_j' = \frac{1}{6h}\left(11z_j - 18z_{j-1} + 9z_{j-2} - 2z_{j-3}\right). \tag{14}$$

Repeat Steps 5 and 6 until changes in z_j' values for two consecutive iterations are small.

An alternative to the above algorithm is Algorithm 2.

ALGORITHM 2.

- Step 1: Set $\alpha = 0$.
- Step 2: Solve (4)–(6) for y_j, z_j, $j = 2, \dots, n+1$ using Newton's Method.
- Step 3: Compute the numerical derivatives z'_1 and z'_{n+1} using, respectively, (7) and

$$z'_{n+1} = \frac{1}{6h}\left(11z_{n+1} - 18z_n + 9z_{n-1} - 2z_{n-2}\right). \tag{15}$$

  Use cubic splines to compute z'_j, $j = 2, 3, \dots, n$. The use of numerical derivatives (7) and (15) for the end points in cubic splines is suggested in [6, p. 56].
- Repeat the Steps 2 and 3 with $\alpha = 10^{-5}$ until the changes in all of the derivatives z'_j, $j = 1, 2, \dots, n+1$ computed in two consecutive iterations become small.

3. COMPUTATIONAL RESULTS

We recall that, for our models [1,2], for each j, the vector $(y_j, z_j) \in \mathbb{R}^{46}$. Therefore, in each deferred-correction iteration, forty-six nonlinear algebraic equations are solved for every level $j = 2, 3, \dots, n+1$ by Newton's method. We will now briefly describe Newton's method (for more details, see [7]).

Let $\gamma = (y_j, z_j)$. Since (y_{j-1}, z_{j-1}) is already known and z'_j and z'_{j-1} are constant (they are functions of variables computed in the previous deferred correction step), equations (4)–(6) can be written as

$$\phi(\gamma) = 0, \tag{16}$$

where $\gamma, \phi \in \mathbb{R}^{46}$.

NEWTON'S METHOD (INNER ITERATION).

Compute an initial guess for γ using known prior values of γ.

- Step 1: Solve

$$\phi'(\gamma)\delta\gamma = -\phi(\gamma) \tag{17}$$

  for $\delta\gamma$. In the above equation, compute (or update) the Jacobian $\phi'(\gamma)$ by a numerical method [7].
- Step 2: Let

$$\gamma \Longleftarrow \gamma + \delta\gamma. \tag{18}$$

- Repeat Steps 1 and 2 until $\|\phi(\gamma)\|_2$ is small.

These Newton iterations are called inner iterations in order to distinguish them from the deferred correction iterations. In all our experiments, the maximum number of these inner iterations was less than six.

We have given the results of some computational experiments in Table 1. We list the relative error in the solutions $S_n^k = (y_2, z_2, \dots, y_{n+1}, z_{n+1})$, where $k = 1$ or 2 (Algorithm 1 or Algorithm 2), n is the number of subintervals into which the x range $(0 \le x \le 0.5)$ is subdivided. We used S_{320}^k as a basis to compare S_{80}^k and S_{160}^k. Let $E_n^k = \|S_n^k - S_{320(n)}^k\| / \|S_{320(n)}^k\|$, where $n = 80$ or 160 and $S_{320(n)}^k$ is a vector (of the same dimension as S_n^k) formed from S_{320}^k by selecting only the elements that correspond to S_n^k. As pointed out in the previous section, the discretization error at each level is $O(h^3)$ and, therefore, the total error after n levels $O(h^2)$, where $h = 0.5/n$. If D_n denotes this discretization error, then we have $D_n = Ch^2 = 0.5 * C/n^2$, where C is some constant, and it follows that $D_{2n}/D_n = 0.25$. From Table 1, we see that the experimental results $E_{160}^1/E_{80}^1 \approx 0.19$ and $E_{160}^2/E_{80}^2 \approx 0.19$ agree with the theoretical estimate D_{2n}/D_n.

In Table 2, we give the relative difference between the two solutions obtained by the two algorithms. Algorithm 1 is a local method viz., the computations are done sequentially (except for

Table 1. Relative errors in the solutions.

	Algorithm 1 $k=1$	Algorithm 2 $k=2$
E_{80}^k	1.16×10^{-4}	1.15×10^{-4}
E_{160}^k	2.17×10^{-5}	2.23×10^{-5}

Table 2. Comparison of solutions from the algorithms.

n	80	160	320
$\frac{\lVert S_n^1 - S_n^2 \rVert}{\lVert S_n^1 \rVert}$	4.17×10^{-6}	2.62×10^{-6}	6.67×10^{-6}

the first four levels). The maximum number of local deferred correction iterations at each level was *three.* Algorithm 2 is a global method, where cubic spline interpolation on z_j; $j = 1, 2, \ldots, n+1$ is used to evaluate the derivatives z_j' and global deferred correction iterations are used to recompute the variables y_j, z_j. The maximum number of such iterations was *ten.* Thus, Algorithm 1 requires significantly less work than Algorithm 2.

REFERENCES

1. A.M. Weinstein, A mathematical model of the rat proximal tubule, *Am. J. Physiol.* **250**, F860–F873 (1986).
2. A.M. Weinstein, Chloride transport in a mathematical model of the rat proximal tubule, *Am. J. Physiol.* **263**, F784–F798 (1992).
3. R.P. Tewarson, H. Wang, J.L. Stephenson and F. Jen, Efficient computer algorithms for kidney modeling, *Math. Modeling & Sci. Computing* 1, 164–171 (1993).
4. H. Wang, J.L. Stephenson, Y. Deng and R.P. Tewarson, An efficient parallel algorithm for solving n-nephron models of the renal inner medulla, *Computers Math. Applic.* **28** (5), 1–12 (1994).
5. Z. Kopal, *Numerical Analysis*, Second edition, John Wiley & Sons Inc., New York, (1961).
6. C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, (1978).
7. R.P. Tewarson, J.L. Stephenson and L.L. Juang, A note on solution of large sparse systems of nonlinear equations, *J. Math. Anal. and Appl.* **63**, 439–445 (1978).